

Zig Bee Communication of Sensor Data Between Raspberry Pi and Ubuntu System

H. Mala¹, R. Chaithra², L. Ranjitha³, G. Sindhu⁴, M. Raghavendra^{5*}

^{1,2,3,4,5*} *Department of Computer Science Engineering, SIT, Tumkur, Karnataka, India.*

Abstract

Raspberry pi (R-pi) is a low cost system on chip (SOC) com-puter which can be operated with various operating systems like Raspbian for ubuntu, Pidora for fedora. This SOC has plenty of GPIO hardware ports through which programmer can access sensors connected to external devices. In traditional computer systems, we can access external hardware through USB/serial ports, where lot of programming and hardware circuits are nec-essary. But Raspberry Pi has built in GPIO pins to which exter-nal sensors are connected and these pins are addressable from python program, using specific library routines. This facility en-ables the designer to directly access the sensor connected to the hardware ports.

Sensors connected to the Raspberry Pi SOC communicates with ubuntu base station using wireless Zigbee technology. In this experiment, live sensor data is transmitted over zigbee chan-nel and received using zigbee receiver and graph is displayed at Ubuntu laptop. Experiments are conducted and graphs are drawn for temperature, humidity, light and motion detection sen-sors.

Keywords: *DHT11; LDR sensor; PIR motion sensor; Python; Raspberry pi; Wireless communication; Zigbee.*

1. INTRODUCTION

Wireless Sensor Networks have intensive applications in IOT (Internet of things) where wireless transmission technologies such as Bluetooth, ZIGBEE, WiFi can be used. Various open hardware devices such as raspberry pi, aurdino can be pro-grammed to perform specific sensing operations. These embed-ded systems uses Python, Java as programming language run-ning in OS like raspbian. The library package facilitates access to digital GPIO and analog IO ports which can be connected to sensors. Each node in WSN will consists of sensor, micro controller and ZIGBEE. The sensed data is fed to the micro con-troller and from there, it will be fed to Zigbee for wireless trans-mission and received by the base station, which can be a laptop connected to internet.

The experiment uses DHT11 Adafruit Library to extract temperature-humidity data and python library for Rpi.GPIO to extract PIR motion sensor, LDR light sensor data.

Experiment uses following Hardware components - DHT-11 temperature-humidity sensor, Raspberry pi board, Zigbee mod-ule, PIR motion sensor, LDR light sensor and UBUNTU laptop.

Following data sheets and manuals are referred to design and program the circuit - PIR [sensor](#)

(http://www.nskelectronics.com/motion_sensor.html), (<https://www.crazyapi.com/buy-grove-pir-motion-sensor?search=motion%20sensor>), light sensor LM393[27],
NSK zigbee cc2500
(http://www.nskelectronics.com/files/cc2500_rf_module.pdf), DHT-11(Adafruit's Raspberry Pi Lesson 4., GPIO Setup, Created by Simon Monk, adafruit learning system) (<https://learn.adafruit.com/dht-humidity-sensing-on-raspberry-pi-with-gdocs-logging/overview>) (https://github.com/adafruit/Adafruit_Python_DHT) and Raspberry pi (<http://sourceforge.net/p/raspberry-gpio-python/wiki/BasicUsage/>).

2. HARDWARE COMPONENTS

2.1 Raspberry pi

The Raspberry Pi-2 Model B used in the experiment is second generation Raspberry Pi (<https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>) SOC, and is more useful for em-bedded projects and projects which require very low power. It is a single board computer for which sensor nodes, Zigbee devices can be connected and it works on Ubuntu like raspbian operating system. The raspberry pi uses the OS deployed at the SD card. Even if linux is not real time OS, the program will be given enough priority and time by the linux kernel (<https://learn.adafruit.com/dht-humidity-sensing-on-raspberry-pi-with-gdocs->

***M. Raghavendra**
email:rag.sit.cse@gmail.com

[logging/overview](#)) to reliably read the sensor. Due to this, there is no guarantee that the program will be given, enough priority and time by the linux kernel to reliably read the sensor. However about 75% of the read requests should generally succeed and return a result.

The raspberry pi can be used along with keyboard, mouse and HDMI monitor (Adafruit's Raspberry Pi, 5). These IO devices can be directly connected to the R-pi board using USB or we can use remote access pro-gram such as putty or remmina desktop client using Ethernet or wifi. We can also use serial pin (tx/rx) to allow devices to connect to its console. Since all of its GPIO pins are digital, they can input-output only high and low levels.

Raspberry pi system is low cost and highly scalable both in terms of the type of sensors and number of sensor nodes [18]. WSN can be built using raspberry pi, which can be low cost, easy to build, and easy to maintain as compared to earlier designs. Also it is low power device, uses 3.3V for its operation applications. The Zigbee can be used as router, co-coordinator and end devices, and has the capability to control the entire network.

ZigBee uses IEEE 802.15.4 standard [10][11], is a communication technology which is suitable for WSN [1], to transmit data from sensor node to base station. The advantage of zigbee is its low cost and low power consumption, which is ideal for resource hungry sensor networks. It is a networking standard [8] for connecting sensors, instrumentation and control systems in a wireless personal area network(WPAN). These technology standards are used in the Internet of things concept where all the above discussed devices are connected to the Internet via WiFi/Ethernet. It has better performance over its competing technologies in the parameters like frequency band, range, current consumption and data rate etc.

2.2 DHT-11

The DHT-11[20] is a basic, ultra low-cost composite sensor, senses both digital temperature and humidity values[17]. It is fairly simple to use, but requires careful timing to grab data having sampling period of 2 seconds. Since it works with DC 3.5-5.5v, the device can be connected directly to GPIO pin of Raspberry pi.

2.3 Zigbee

The circuit uses cc2500 ZigBee [13], and is used in the wireless communication. It is connected to raspberry pi using USB port [15] at the sender side and UBUNTU laptop at the receiver side. A pair of ZigBee of the same series needs to be used in order to make the communication. Sensor networks which operate on battery power[3], has a low data rate of 250kbps[2] and radio range of 20m to 100m[8], which is higher value compared

to competitive technologies like Bluetooth and wifi. CC2500 is a plug and play device, no external antenna and no configuration is needed and its range is max 30m (line of sight). This module works in serial communication and provides bidirectional, but only one direction at the same time(half-duplex), operates at frequency of 2.4 GHz, transmit and receive data at baud rate of 9600, and are particularly suitable for fast prototyping for WSN

2.4 Motion Sensor - PIR

Motion sensor is a Passive Infra Red (PIR) sensor, which will detect motion of objects passing in front of it.

2.5 Light Sensor - LM-383

Light Dependent Resistors (LDR) is a variable resistor device, allows to measure a number of light levels using a single GPIO pin.

3. PROGRAMMING THE HARDWARE USING PYTHON

3.1 Python and Raspberry pi

The single board computer Raspberry Pi [19] uses python as one of its programming language. It is an easy to use programming language and lets the user to connect Raspberry Pi to the real world[21]. A Python program contains a series of commands that the computer executes from top to bottom. It is a versatile high level language, with easy to read syntax that allows programmers to use fewer lines of code than would be possible in language such as assembly, c or java[22].

Using with Raspberry Pi, python lets the user to connect the project to the real world. It has the facility to access GPIO pins of Raspberry, through programming instructions, which will simplify the interface, which will allow access to these pins.

Python has GPIO library [23], which has APIs to perform operations on hardware modules connected to GPIO pins. Python GPIO library has the facility to access GPIO pins of R-pi, through which user connects his project to the real world. Raspberry pi has many exposed interfaces and power full CPU makes flexible base platform to evaluate sensor devices, and prototype data collection and compute applications[24]. We can connect simple sensors to GPIO pins and for complex sensors we can use the USB. The GPIO set up is required for raspberry pi [12]. To program the GPIO ports in python, we need to install a very useful python2 library called Rpi.GPIO. Instructions for installing the library can be found at raspberrypi.org¹.

3.2 DHT-11 And Adafruit library

Adafruit developed python library for DHT sensor[5][20], and for many other sensors. Raspberry pi

controlled wireless sensor networks, can be used in traffic monitoring [3] systems. For wireless networking we can use Zigbee[2], Bluetooth[4], wifi[8] and GPRS. Humidity-temperature sensors libraries[9][16] are downloaded from adafruit².

4. DESIGN AND EXPERIMENTAL SET UP

The execution sequence is as given below.

1. Receiver: (UBUNTU)

```
sudo python recv.py
```

2. Sender: (raspberry pi)

```
sudo python sender.py
```

In any client-server type of environment, the server must be ready to receive client requests. now the client can initiate its communication. In the current experiment also, the receiver acts server and must started early. Sender R-pi will sense the sensor data and send it to the base station.

4.1 Design of sensor node circuit diagram

DHT-11 sensor is connected to raspberry pi, which senses the temperature and humidity data and it is fed to the python pro-gram using GPIO pin 21(h/w pin number 40), with a pull up resistor 4.7K or 10K (ohm) as shown in the fig.1. This resistor works as current limiter and as load balancer, which will be used as a pull-up resistor from the data pin to VCC. The data is fed into the ZIGBEE module, connected through USB port of raspberry pi.

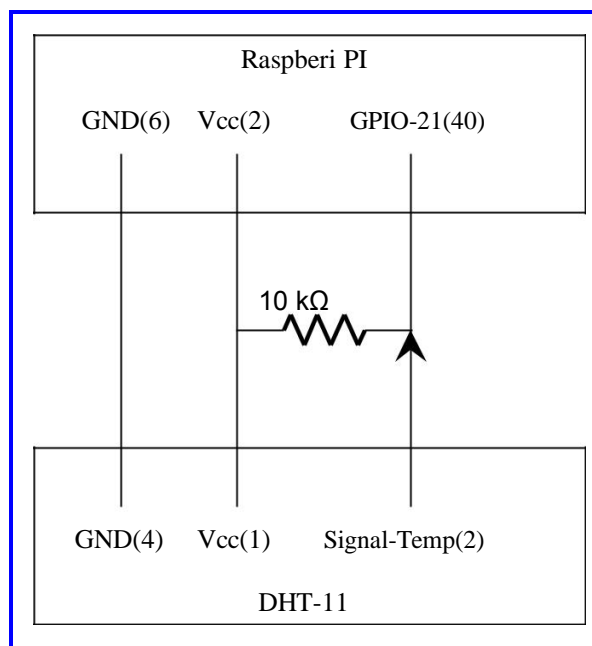


Fig. 1: Raspberry pi connected to DHT-11 Sensor

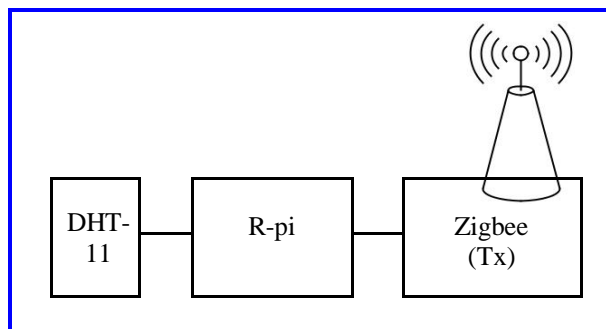


Fig. 2: Block Diagram of the DHT-11 (Sender)

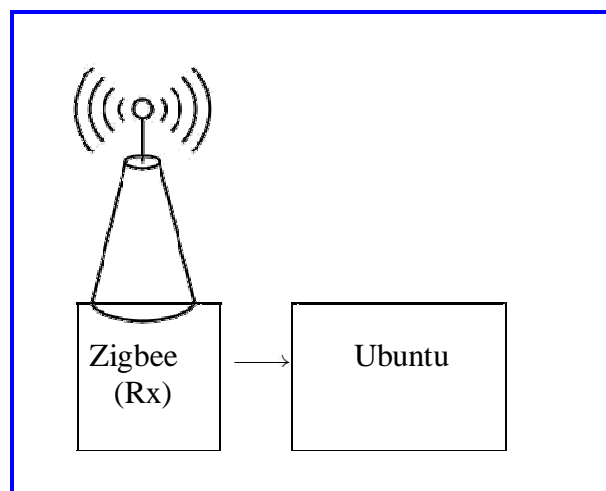


Fig. 3: Block Diagram of the DHT-11 (Reciever)

Similarly, other sensors are also connected and programmed using R-pi. The pin assignment in the circuit is as shown the table.1. The python code calls DHT library which in turn reads temperature and it is returned back to the python program as shown in the state diagram at Fig.4. in which Python Code, calls Adafruit-DHT Library.

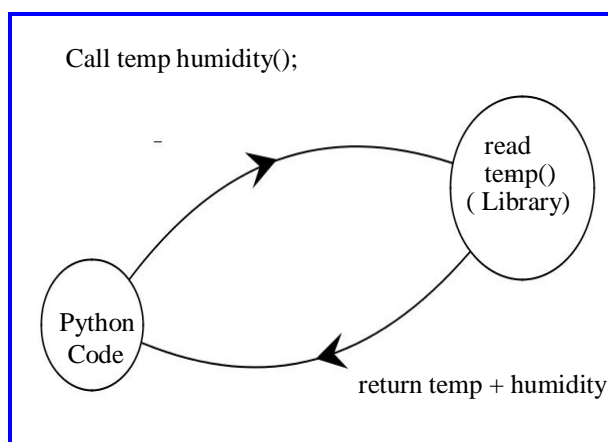


Fig. 4: State Diagram of reading temperature, humidity data from DHT-11

Table 1.

Raspberry pi side HW (GPIO)	signal name	Sensor name /pin no
6	GND	DHT-11 /4
2	VCC	DHT-11 /1
40(GPIO-21)	signal-Temp	DHT-11 /2
6	GND	PIR / GND
2	VCC	PIR / VCC
26 (GPIO-7)	signal-motion detected	PIR / OUT
6	GND	LM-393 / GND
2	VCC	LM-393 / VCC
29(GPIO-5)	signal-light-intensity	LM-393 / A0

1. Algorithm: Transfer sensor data using Zig Bee

Algorithm zig bee sender()

```

- -
1. [import following libraries]
Serial, time,adafruit DHT,
matplotlib, RPi.GPIO
2. [initial raspberry pi serial port]
Ser=serial.serial('/dev/USB0',9600);
3.[Set GPIO mode to BCM.
4.Use following R-pi GPIO pin
configuration] temp,humidity GPIO-21
pir GPIO-7
LDR GPIO-5
5.[repeat step 5 to step 10 until
termination send the sensor data to
zigbee]
6.[read temperature(temp), humidity
using DHT library]
Humidity ,temperature= Adafruit DHT.
read retry(sensor,pin)
7.[write temperature into - serial
port]
Ser.write(temp)
8.[write humidity into serial port]
Ser.write(humidity)
9.[ Read PIR Status (pir) and write
into serial port ] pir=read(PIR);
Ser.write(pir);
10. [ Read light intensity (ldr ) and
write into serial port ] ldr =
read(LDR)
Ser.write(ldr)
11.End of sender algorithm.
```

Algorithm zig_bee receiver()

```

1. [import following python
package] Serial, time sys, matplotlib
2. [Initialise serial port]
Port=serial.serial("/dev/ttyusb0")
3. [repeat step 3 and step 11
until termination
Read incomming sensor data using
zigbee and display it on terminal]
4. Rcv=port.readline
5. Print "temp= ",Rcv
6. Rcv=port.readline
7. Print "humidity= ",Rcv
8. Rcv=port.readline
9. Print "PIR= ",Rcv
10. Rcv=port.readline
11. Print "Light= ",Rcv
12. End of receiver algorithm.
```

4.2 Programming Sender side (Raspberry pi)

Following code will read temperature data from DHT-11 and write it to zigbee connected to USB port.

Following python library packages needs to be included.

```

import serial
import time
import
datetime
import numpy
as np import
sys
import
Adafruit_DHT
import RPi.GPIO as
GPIO import time
```

Configure GPIO pins for read sensor data and write to USB serial port for zigbee transmission

```

ser=serial.Serial('/dev/ttyUSB0' ,
9600)
GPIO.setmode(GPIO.BCM)
pin = 21
GPIO.setup(pin,GPIO.IN)
```

The sensor data contains temperature and humidity values and are needed to be separated. The DHT-11 adafruit python library will split temperature and humidity values.

```

try:
while True:
humidity, temperature = Adafruit_DHT.
read_retry(sensor, pin)
ser.write(temperature)
time.sleep(1)

```

4.3 Programming base station / Receiver (Ubuntu)

The UBUNTU laptop is used as the receiver and base station in the communication. The receiver ZigBee is connected to the USB port of the laptop. The received data will be displayed on the terminal.

Following python library packages needs to be included.

```

import serial
import time
import datetime
import numpy as np
import sys

```

For plotting graphs, we need to include matplotlib.

```

from matplotlib import pyplot as plt
port=serial.Serial("/dev/ttyUSB0",
baudrate=9600, timeout=300.0)

```

initialization for graphs

```

ymin=0
ymax=60
plt.ion() # set plot to animated
ydata = 0

```

Graph is plotted by taking time along x-axis and room temperature along y-axis

```

line, = plt.plot(ydata)
plt.ylim([0,4])
plt.xlabel('Time ( Sec ) ')
plt.ylabel('Temp ( Deg. Cel. )')
plt.title('Room Temperature ')

```

The live graph is plotted using the data obtained from USB serial port by using zigbee receiver. Following code will read live temperature data from the zigbee receiver.

```

while True:
plt.ylim([ymin,ym
ax]) rcv =
port.readline()

```

```

print("\r\nat recvr
temp:" + repr(rcv))
rcv=repr(rcv))
ydata.append
(rcv) del
ydata[0]
line.set_xda
ta
(np.arange(len(ydat
a)))
line.set_ydata(yda
ta) plt.draw()

```

4.3.1 PIR Motion Sensor

The data received from PIR sensor is transmitted using follow-ing code, and the status is transmitted in per second basis.

```

sender
:
PIR_PI
N = 7

GPIO.setup(PIR_PIN,
GPIO.IN) while True:

if GPIO.input(PIR_PIN):
print "Motion Detected!"
ser.write("Motion
Detected!")
else

ser.write("Motion Not
Detected So
far")

sleep(1)

```

At the ubuntu side, motion sensor data is received and motion detection status is printed at terminal and displayed using graph.

```

receiver :
rcv = ser.readline()
print Status :
print rcv

```

4.3.2 LDR Light Intensity Sensor

The LDR sensor is connected the GPIO-5 pin of R-pi. And the incoming data is fed to zigbee at the sender side.

```

sender:
sensor_value = GPIO.input(5)
ser.write("\n light ");

```

```
str5 = str(sensor_value)
ser.write(str5)
```

The live data is converted into percentage value.
receiver:

```
rcv = port.readline()
ldr=float(rcv)
if ldr > 0.0 :
    per_ldr = (float) (ldr* 100 ) /
    1024 print per_ldr
```

5. EXPERIMENTAL RESULT

Graphs are plotted using python library matplotlib, between time in seconds along X-axis versus sensor data along Y-axis. Fig.5 shows the live room temperature at the time of the experiment measured in degree celcius. Fig.6 shows humidity statistics calculated in terms of percentage. Fig.7 has light sensor output. The LDR sensor senses the light intensity values, displayed in per-centage. Fig.8 shows motion detection pulse triggered by passing of an object. When the motion is detected by the LDR sensor, a pulse is triggered at the graphical output.

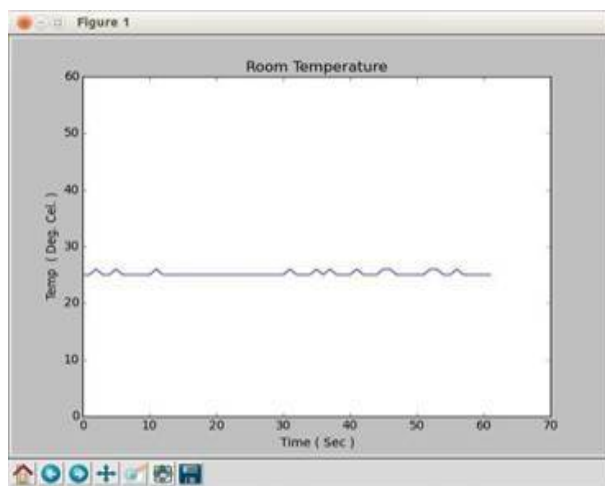


Fig. 5: Temperature Graph

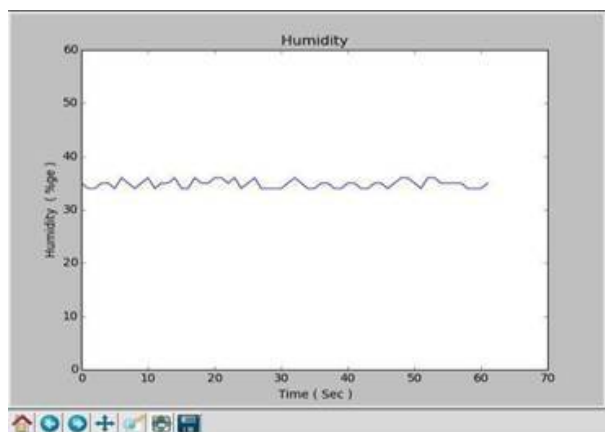


Fig. 6: Humidity Graph

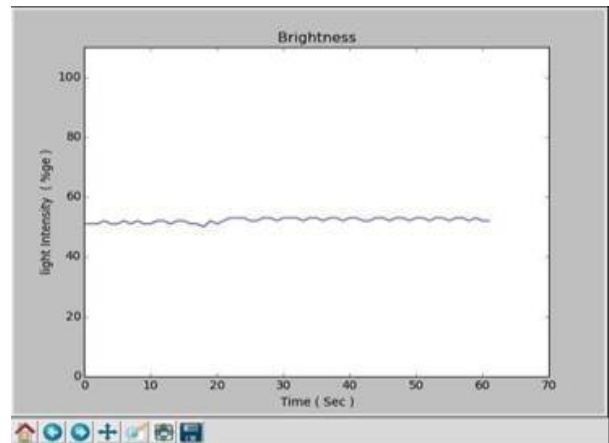


Fig. 7: Output of light Sensor

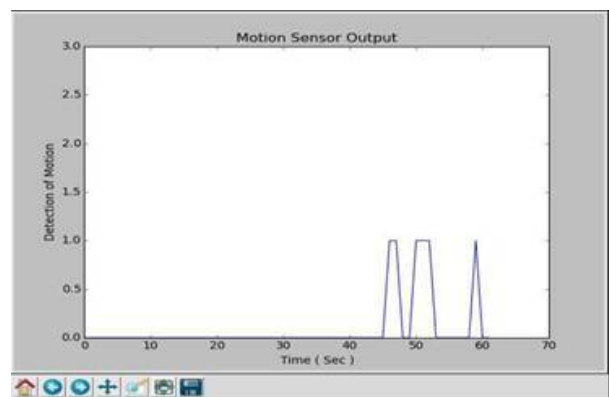


Fig. 8: Motion Detection Graph

6. CONCLUSION

The live sensor data is communicated from raspberry pi to UBUNTU system using ZIGBEE technology. The experiment uses temperature, humidity sensor, motion sensor and light sensor. Various packages are available in Python language in order to interface with sensor devices connected to GPIO pins of Raspberry Pi. The experimental results are displayed at ubuntu receiver both by using terminal output and graphs, plotted by using matlab library for python. Live data of room temperature and humidity is measured using DHT-11 sensor. PIR sensor senses motion of moving objects. Light intensity is measured using LDR sensor. The future work can be done to develop real time system.

REFERENCES

- [1] Meng-Shiuan Pan and Yu-Chee Tseng, "Communication Protocols and Applications for ZigBee-Based Wireless Sensor Networks", Taiwan-French Conf. on Information, 2006, Web : people.cs.nctu.edu.tw
- [2] Rohitha, P., Ranjeet Kumar, P., Adinarayana, N. and Venkat Narayana Rao, T., Wireless

- Networking Through ZigBee Technology, International Journal of Advanced Research in Computer Science and Software Engineering, 2(7), (2012) online at: www.ijarcse.com
- [3] Michal Kochla, Michal Hodon, Lukas Cechovi, "WSN for Traffic Monitoring using Raspberry Pi Board", Pro-ceedings of the 2014 Federated Conference on Computer Science and Information Systems, 2, 10231026, doi: 10.15439/2014F310AC SIS.
 - [4] Gopinath Shanmuga Sundaram, Bhanuprasad Patiban-dala, Harish Santhanam, Sindhura Gaddam, Vamsi Kr-ishna Alla, Gautham Ravi Prakash, Shiva Chaitanya Vishwakarma Chandracha, Sindhu Boppana and James M. Conrad, "Bluetooth Communication using a Touch-screen Interface with the Raspberry Pi", Southeastcon, Jacksonville, FL, 4-7 April 2013 , Proceedings of IEEE , pages 1 - 4 , DOI:10.1109/SECON.2013.6567448.
 - [5] Adafruit Python DHT Sensor Library , <https://github.com/adafruit/AdafruitPythonDHT>
 - [6] DHT Humidity Sensing on Raspberry Pi or Beaglebone Black with GDocs Logging , <https://learn.adafruit.com/dht-humidity-sensing-on-raspberry-pi-with-gdocs-logging/overview>
 - [7] DHT-xx Sensors - Overview , <https://learn.adafruit.com/dht/overview>
 - [8] Ankur Tomar, "Introduction to Zibgbee Technology", Global Technology Centre , Volume 1, July 2011
 - [9] DHT Humidity Sensing on Raspberry Pi or Beaglebone Black with GDocs Logging , Created by lady ada, adafruit learning system, <https://learn.adafruit.com/dht-humidity-sensing-on-raspberry-pi-with-gdocs-logging/overview>
 - [10] Meng-Shiuan Pan and Yu-Chee Tseng, "ZigBee Wireless Sensor Networks and Their Applications"
 - [11] Akiba, "Zigbee Network Layer Tutorial- PartI: The TxData Path, <http://freaklabs.org/index.php/Blog/Zigbee/Zigbee-Network-Layer-Tutorial-Part-1.html>, (2009), Network Tutorial
 - [12] Adafruit's Raspberry Pi Lesson 4., GPIO Setup, Created by Simon Monk, adafruit learning system
 - [13] CC2500 RF Module - Zig Bee Data Sheet , http://www.nskelectronics.com/files/cc2500_rf_module.pdf
 - [14] Adafruit's Raspberry Pi Lesson 5., Using a Console Cable Created by Simon Monk, adafruit learning system, <https://learn.adafruit.com/adafruits-raspberry-pi-lesson-5-using-a-console-cable/overview>
 - [15] Prodigy Productions, LLC Add XBee to Raspberry Pi with Python, July 17, 2013, in Programming, Python, Raspberry Pi, Software Development, by LuCuS
 - [16] DHT Humidity Sensing on Raspberry Pi or Beaglebone Black with GDocs Logging, Created by lady ada
 - [17] Temperature and humidity module DHT11 Product Manual, www.aosong.com
 - [18] Sheikh Ferdoush, Xinrong Li, Wireless sensor network system design using raspberry Pi and Arduino for environmental monitoring applications, Procedia Computer Science, 34, 103-110(2014).
 - [19] RASPBERRY PI 2 MODEL B" , <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>
 - [20] DHT11 BASIC TEMPERATURE-HUMIDITY SEN-SOR, <https://www.adafruit.com/product/386>
 - [21] RASPBERRY PI Python-blog, <https://www.raspberrypi.org/documentation/usage/python/>
 - [22] How to Write and Run a Python Program on the Rasp-berry Pi, <http://www.circuitbasics.com/how-to-write-and-run-a-python-program-on-the-raspberry-pi/>
 - [23] Rasperry-GPIO-Python – A Python module to control the GPIO on a Raspberry Pi, <http://sourceforge.net/p/raspberry-gpio-python/wiki/BasicUsage/>
 - [24] Kristian Hentschel, Dejice Jacob, Jeremy Singer and Matthew Chalmers, Supersensors: Raspberry Pi Devices for smart campus infrastructure, IEEE 4th International conference on future internet of things and cloud, 2016.
 - [25] Grove-PIR Motion sensor, <http://www.crazyapi.com/buy-grove-pir-motion-sensor?search=motion%20sensor>
 - [26] http://www.nskelectronics.com/motion_sensor.html
 - [27] <http://www.alldatasheet.com/view.jsp?Searchword=LM303->

